UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/788,801	02/27/2004	Scott Musson	BEAS-01382US0	7864

23910 7590 07/05/2007
FLIESLER MEYER LLP
650 CALIFORNIA STREET
14TH FLOOR
SAN FRANCISCO, CA 94108

EXAMINER	
HASSAN, RASHEDUL	
ART UNIT	PAPER NUMBER
2179	

MAIL DATE	DELIVERY MODE
07/05/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/788,801	**Applicant(s)** MUSSON ET AL.
	Examiner Rashedul Hassan	**Art Unit** 2179

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☐ Claim(s) *1-45* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☐ Claim(s) *1-45* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Specification

Claims 30-44 are directed to a machine readable medium. Regarding a "machine readable medium", the disclosure only mentions, "Stored on any one of the computer readable medium (media), the present invention includes software for controlling both the hardware of the general purpose/specialized computer or microprocessor, and for enabling the computer or microprocessor to interact with a human user or other mechanism utilizing the results of the present invention" [0116]. As such, the disclosure fails to adequately disclose as to what constitutes the claimed machine readable medium. In the absence of any evidence of applicant's intent to the contrary, the reasonable interpretation of a machine readable medium conveyed to one of ordinary skill in the art is appropriate tangible physical article or objects under the meaning of 35 U.S.C 101 and the machine readable medium recited in claims 30-44 have been interpreted likewise. However, the applicant is encouraged to correct this deficiency in the disclosure.

Claim Objections

Claims 30 and 45 are objected to because of the following informalities:

Each claim should begin with a capital letter and ends with a period (see MPEP 608.01(m) Form of Claims [R-3]). Claims 30 and 45 do not end with a "period". Appropriate correction is required.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claim 45 is rejected under 35 U.S.C 101 for being directed to non-statutory subject matter.

Claim 45 is directed to a computer data signal embodied in a transmission medium. A signal is currently deemed to be a non-statutory subject matter under the meaning of 35 U.S.C 101.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-13, 15-27, 29-42, and 44-45 are rejected under 35 U.S.C. 102(b) as being anticipated by Anuff et al. (US 6,327,628 B1).

For claims 1, 16, 30 and 45, Anuff anticipates a computer implemented method for responding to a request (an example of a request is disclosed in [0039] of the instant specification to be "an HTML request originating from a web browser. Anuff teaches this limitation. See c3:13-22, c6:57-65, also c14:32-36) , comprising:

accepting the request (an example of this step is disclosed in [0039] of the instant specification, wherein a process such as a web/application server or other suitable process can accept the request. Anuff teaches this limitation. See Fig. 1, c3:1-24) ;

mapping the request to a control tree wherein the control tree is a logical representation of a graphical user interface (GUI) and wherein the control tree includes a set of controls which are related hierarchically to one another (A "control tree" is recited in the claim to be "a logical representation of a graphical user interface" containing a set of hierarchically related controls. According to the disclosure, "controls" represent "corresponding graphical and functional elements in web applications ... In one embodiment, a control can be implemented as one or more classes in an object oriented programming paradigm" [0028]. A "logical representation" is nothing more than an abstract idea for conceptually viewing the GUI as a set of related back end processing components. These components can be, but not necessarily, implemented as objects instantiated from classes in an object-oriented programming paradigm (hereinafter referred to as OOP). Therefore, a "control tree" can reasonably be interpreted to mean, the GUI itself or alternatively, all the relevant back end processing components/objects implementing the requested GUI; see [0036]. Similarly, "mapping the request to a control tree" can reasonably be interpreted to mean, identifying the entire relevant back end processing components/objects implementing the requested GUI. Anuff inherently teaches identifying the entire relevant back end processing components/objects implementing the requested GUI. He further teaches, with regard to Fig. 4, that these back end controls/objects are related hierarchically to one another, e.g., A owns B and A is a subclass of B). ;

advancing the control tree (e.g., all the back end processing components/objects implementing the GUI) through at least one lifecycle stage based on the request(For a control, the lifecycle is defined in the instant disclosure, by a set of methods representing stages in the lifecycle. Life cycle stages are illustrated in Table 3 and appear to be nothing more than various stages of an object, instantiated from a class in the context of OOP, during runtime. Therefore, a GUI implemented using objects in OOP, inherently advances the objects implementing the GUI through at least one lifecycle stage, e.g., at least the "Init" stage that allows a control to perform initialization. This stage is usually implemented by way of using a "constructor" function to instantiate an object. Thus Anuff clearly teaches this limitation since his implementation is based on OOP paradigm), wherein the control tree includes at least one portlet control that represents at least one portlet (the instant disclosure shows examples of "portlets" as elements 18, 30 and 32 in Fig. 1 and states, "A portlet is an application that manages its own GUI ...By way of a non-limiting example, portlet 30 displays real-time stock ticker information. A user could configure such a portlet to display certain stocks, for example. In another embodiment, the user can select a given stock displayed in portlet 30 and receive more detailed information, such as the price history, price to earnings ratio, etc." [0026]. In short, "Portlets" are generally known in the art to be pluggable user interface components that are managed and displayed in a web portal. It resembles a web-based application module that is hosted in a Portal. Anuff teaches use of at least one portlet control that represents at least one portlet, see modules 26 in Fig. 2);

providing the request to a portlet container that contains the at least one portlet (the instant disclosure mentions "In a framework, controls can also serve as containers for other controls. By way of a non-limiting example, a page may contain a booklet and a portlet ... " [0028]. The instant disclosure presents Fig. 2 as an illustration of a control taxonomy in accordance to an embodiment, wherein control/object like a web application 200, portal 202, page 218 etc. containing a portlet object functions as a "portlet container". Anuff teaches, as shown in Fig. 2, providing the HTTP request from a browser to a server processes 12a-12n that serve as portlet containers); and

aggregating the output of each of the at least one portlets and providing the output to the GUI (in this context, "providing the output to the GUI", is interpreted to mean rendering the output on the display device. Anuff clearly teaches this limitation as shown in Fig. 2).

For claims 2, 17 and 31, Anuff further anticipates generating the control tree (the "control tree" is interpreted to mean the GUI itself or alternatively, all the relevant back end processing components/objects implementing the requested GUI) from a factory based on the request (the instant disclosure says about the "factory", with regard to Fig. 3, "In one embodiment, a GUI is rendered in response to a request (e.g., an HTML request originating from a web browser). In step 300, a process (e.g., a web/application server or other suitable process) can accept the request and map it to a **control tree factory**. In step 302, the identified **factory** can be used to generate a control tree representing the GUI" [0040]. With regard to Fig. 4, the disclosure says about the factory, "A request 410 can be mapped to a control tree factory 402 by container 400. In one embodiment, the container use "wire-up" service 404 in the factory which will cause the factory to return the root of a control tree. In one embodiment, the generation of the control tree is based in part on the request. The control tree (not shown) produced by control tree factory can be a page level object or can become a sub-tree of a larger control tree. The control tree factory is independent of container 400 and can be accessed from multiple containers. In one embodiment, the control tree factory can make modifications to the tree such as replacing the default rendering or state methods for a control, including for the page itself". Fig. 5, Fig. 6 and Fig. 7 diagram a control tree factory having a JSP page description implementation, a metadata page description implementation and a pure JSP page description implementation respectively. Based on the descriptions regarding the figures, it appears that a control tree factory can be interpreted to mean, in absence of any explicit definition of the term "factory" in the disclosure and without importing limitations from the disclosure into the claims, to be objects/processes arbitrarily combined or divided into separate software, firmware or hardware components needed to create the GUI [0046-0052]. Therefore, any of the servers 12a-12n taught by Anuff can be interpreted as the "factory").

For claim 3 and 32, Anuff further anticipates generating a response wherein the response can be used to render at least a portion of the GUI (since the response from servers 12a-12n are used to display modules 26 in portal front page. These modules are objects that encapsulate a specific, bounded portion of the GUI, and allow that portion to be administered as a unit. For example, a module might display news, sports scores, stock quotes, or weather forecasts, c3:2-24 and c6:22-31).

For claim 4, 18 and 33, Anuff further anticipates that the step of generating a control tree from the factory comprises: creating a metadata representation (regarding a "metadata representation" the instant disclosure says, "In one embodiment, the metadata representation can be an XML document or Java class file defined by a schema".) of a control tree; and generating a class to construct the control tree based on the metadata representation (Anuff: c6:34-46).

For claim 5, 19 and 34, Anuff further anticipates that the request is a hypertext transfer protocol request (HTTP) (c6:57-58) and the request originates from a web browser (16 in Fig. 1).

For claim 6, 20 and 35, Anuff further anticipates providing the response to a web browser (Fig. 1, Fig. 2, c13:53-55).

For claim 7, 21 and 36, Anuff further anticipates that the control tree is advanced through the at least one lifecycle stage by an interchangeable lifecycle component (regarding an "interchangeable lifecycle component" the disclosure mentions, in regard to Fig. 8, "The control container can use an interchangeable lifecycle driver 804 to drive the control tree through a sequence of states so that the request can be processed. As with the interchangeable persistence driver, an interface is provided to isolate lifecycle driver implementation details from the control container. This allows for different lifecycle implementations to be interchanged as needed". As for what constitutes the "interchangeable lifecycle driver/component", a reasonable interpretation would be, in absence of any explicit definition of the term in the disclosure and without importing limitations from the disclosure into the claim, to be objects/processes arbitrarily combined or divided into separate software, firmware or hardware components responsible to instantiate and carry out the run-time processing of the relevant back end processing components/objects implementing the requested GUI which is inherent in Anuff).

For claim 8, 22 and 37, Anuff further anticipates that each one of the set of controls can have an interchangeable persistence mechanism (regarding an "interchangeable persistence mechanism" the instant disclosure mentions, in regard to Fig. 8, "Controls in the control tree can make use of a persistence interface that acts as a front-end to an interchangeable persistence driver 806. The persistence interface hides persistence implementation details from controls and allows for a flexible architecture where different persistence providers can be "plugged in" as needed" [0056]. Disclosure also mentions, "Controls have the ability to persist state across HTTP (Hypertext Transfer Protocol) requests. A state management API can be provided to give each control in the tree the ability to persist itself before rendering an HTTP response. When an HTTP submit to the same page is received, this saved state can be used to re-hydrate or restore the control tree from its persisted state. Thus, the same state can be maintained across different instances of the same control tree with minimal effort to the control author. Controls can be persisted using a state management persistence mechanism" [0057]. Anuff teaches object persistence using suitable database interface. See c4:16:32 and c5:45-48)

For claim 9, 23 and 38, Anuff further anticipates that each one of the set of controls can render itself according to a theme (c8: 22-49).

For claim 10, 24 and 39, Anuff further anticipates that each one of the set of controls can interact with another one of the set of controls (c4: 60-61).

For claim 11, 25 and 40, Anuff further anticipates that one of the set of controls can advance through the series of at least one lifecycle stage in parallel with another of the controls (since in OOP, objects can be instantiated in parallel and individually carry on their run-time processing in parallel with another object. Anuff also teaches multithreaded module preparation, c14:31-41).

For claim 12, 26 and 41, Anuff further teaches that a lifecycle stage is one of: init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose (implicitly taught since objects apparently follow these stages in OOP which is well known to a person of ordinary skill in the art).

For claim 13, 27 and 42, Anuff further anticipates that the response is an hypertext transfer protocol (HTTP) response (c6:61-65).

For claim 15, 29 and 44, Anuff further anticipates that each one of the set of controls can be one of: Book, Page (c4:65), Window, Menu, Layout (c4:66), Portlet (modules, c4:65), Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claim 14, 28 and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Anuff.

For claim 14, 28, and 43, Anuff does not explicitly teach that controls can raise events and respond to events. However, he explicitly teaches that an object model comprises a collection of objects that work together in documented relationships. Official notice is taken that in object oriented programming communication/co-operation between objects using events was well known in the art at the time of the invention. Therefore, if not already implicitly taught by Anuff, it would have been obvious to a person of ordinary skill in the art to modify his invention so that controls can raise events and respond to events. The motivation for such modification would have been necessitated by the very nature of the GUI (portal) which is an interactive application and it is well known to a person of ordinary skill in the art that such applications are well suited for an event-driven implementation.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Rashedul Hassan whose telephone number is 571-272-9481. The examiner can normally be reached on M-F 7:30AM - 4PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on 571-272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

(Rashedul Hassan)

WEILUN LO
SUPERVISORY PATENT EXAMINER